UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

Universitat Politècnica de Catalunya
Escola Superior d'Enginyeria Industrial, Aeroespacial i
Audiovisual de Terrassa

Realitzat per: Marta Barrachina Català

IDENTIFICACIÓ NO-SUPERVISADA DE PERSONES EN PROGRAMES DE TELEVISIÓ

Supervisor: Josep Ramon Morros Rubió

Terrassa, Octubre 2018

Abstract :

Recently, the amount of video data and images has increased, creating several annotation and classification problems of the data set.
One of these problems was the identification of persons in videos, that's why; lately the research in this area has become more popular.

The aim of this project is to find a new algorithm in order to improve the unsupervised identification of persons in a sequence of a video for TV programs. So as to implement this enhancement, a new classifier will be created from zero to identify if a person in a video sequence is talking or not.

This classifier will be created extracting the person faces in different videos, and classifying it by hand. From here, with the amount of date obtained, mouth will be detected, and the distance between the lips will be measured as well, in order to create a vector of distances for each face detected.

Moreover, an improved facial detector will be applied and the results will be compared with an older detector.

Key words: face recocongition, face detection, clasifier, vídeos annotation and landmarks detection.

Resumen:

En los últimos años la cantidad de datos de vídeos e imágenes ha ido aumentando, esto ha provocado diferentes problemas de anotación y clasificación del conjunto de datos.
Uno de estos grandes problemas es la identificación de personas en videos, es por eso que el estudio en este ámbito ha incrementado en los últimos años.

El objetivo de este proyecto es encontrar un nuevo algoritmo para poder mejorar la identificación no supervisada de personas en secuencias de video para programas de televisión. Para llevar a cabo esta mejora se creará un nuevo clasificador desde cero para poder identificar si una persona en una secuencia de video esta hablando o no.

Este clasificador será creado a partir de la extracción de las caras de personas en diferentes videos, y clasificándolos manualmente según si hablan o no. A partir de este conjunto de datos, se detectarán las bocas y se mesurará la distancia entre los labios, para poder crear un vector de distancias para cada cara detectada.

Además, se aplicará un detector facial mejorada y se compararán los resultados con otro detector mas antiguo.

Finalmente, se exponen los resultados una vez aplicado el nuevo clasificador.

Palabras en clave: reconocimiento facial, detección facial, clasificador, anotación de vídeos y detección de landmarks

Resum:

En els darrers anys la quantitat de dades de vídeos i imatges ha anat augmentant, això ha provocat diversos problemes d'anotació i classificació del conjunt de dades.
Un d'aquests grans problemes és la identificació de persones en vídeos, és per això que la recerca en aquest àmbit ha incrementat en els últims anys.

L'objectiu d'aquest projecte és trobar un nou algoritme per poder millorar la identificació no supervisada de persones en seqüències de vídeo per programes de televisió. Per poder dur a terme aquesta millora es crearà un nou classificador des de cero per poder identificar si una persona en una seqüència de vídeo està parlant o no.

Aquest classificador serà creat a partir de l'extracció de les cares de persones en diferents vídeos, i classificant-les manualment respecte si estan parlant o no. A partir d'aquest conjunt de dades, es detectaran les boques i es mesurarà la distància entre els llavis, per tal de poder crear un vector de distàncies per cada cara detectada.

A més a més, s'aplicarà un detector facial millorat i es compararan els resultats amb un altre detector més antic.

Finalment s'exposaran els resultats un cop aplicat el nou classificador.

Paraules clau: reconeixement facial, detecció facial, classificador, anotació de vídeos i detecció de landmarks.

Agraïments:

En primer lloc, agrair al tutor d'aquest projecte, Josep Ramon Morros, per ajudar-me i guiar-me en aquest projecte. També a l'Albert Gil pel suport que m'ha donat en l'ús del servidor.

Agrair a l'Arnau Folch per l'ajuda proporcionada durant tot el transcurs del projecte.

Finalment agrair a tota la meva família, parella i amics que m'han recolzat i animat durant tots els anys de la carrera. Gràcies per ajudar-me a arribar on sóc i per no perdre mai la confiança en mi.

Moltes gràcies a tots.

Índex

Llistat de figures

Llistat de taules

1. Introducció

El reconeixement de persones en els darrers anys ha esdevingut una àrea de recerca molt activa, ja que té un abast molt ampli en diferents sectors: control de seguretat, seguiment de persones, reconeixement de patrons, emocions etc.

En concret, ens centrem en l'anotació no supervisada de persones en una seqüència de vídeo, que es defineix com a la indexació automàtica de programes de televisió. Així doncs, poder identificar a les persones que apareixen en el transcurs d'un vídeo sense la necessitat de tenir una base de dades d'aquestes entrenada prèviament.

La UPC (Universitat Politècnica de Catalunya) l'any 2015 i 2016 va participar en el projecte europeu *Camomile* (2012-2016) que es basa en millorar i plantejar nous algoritmes d'anotació. Aquest projecte va participar a les diferents avaluacions de *MediaEval* [1]. Aquesta és una iniciativa dedicada a avaluar nous algoritmes d'accés i recuperació multimèdia. El projecte més recent que la UPC va implementar en aquesta iniciativa va ser *Person Discovery in Broadcast TV Task*, tal i com es pot veure en el següent enllaç [2] [3].

Tanmateix, aquest projecte es basa en millorar els mètodes plantejats actuals [3]. Per poder aconseguir aquesta millora, s'aprofiten els recursos que el propi vídeo ens proporciona, ja sigui text, seqüència de característiques facials o el propi àudio del vídeo. En el meu cas, he aprofitat les característiques de la cara, en concret, la boca.

1.1 Objectius

El principal objectiu d'aquest projecte és millorar el sistema no supervisat de vídeos en programes de televisió, explorant nous mètodes per a la classificació, detecció y reconeixement facial.

En aquest treball s'investiga la opció de poder classificar qui està parlant a partir de les característiques de la boca, així, juntament amb el text o amb les característiques que ens proporciona l'àudio, poder etiquetar la persona amb el nom corresponent

Aquest classificador es crearà a partir de diferents vídeos on apareixen varies persones, a continuació s'extrauran les cares detectades i s'anotaran manualment, depenent si la persona parla o no parla. Finalment, s'avaluarà el nou classificador.

1.2 Requeriments

- Aplicar un detector facial millorat per l'extracció de cares d'un vídeo.
- Crear manualment un classificador binari de cares.
- Aplicar i avaluar aquest classificador .
- Utilitzar el llenguatge de programació *Python* per desenvolupar el classificador i aplicar el detector.

1.3 Pla de treball

El treball s'ha implementat de la següent manera:

W1	Introducció a *Pytorch* • Realització de *Fine-tunning*	14/02/18	03/04/18
W2	Implementació del detector facial	03/04/18	15/05/18

W3	Revisió de l'estat de l'art i estructuració del projecte	15/05/18	22/05/18
W4	Creació de la base de dades	22/05/18	17/07/18
W5	Creació del classificador	5/06/18	03/09/19
W6	Correcció d'errors	03/09/18	12/09/18
W7	Realització dels experiments i avaluacions	12/09/18	30/09/18
W8	Redacció del treball	19/09/18	03/10/18

Taula 1. Pla de treball

A continuació es mostra el diagrama de Gantt:

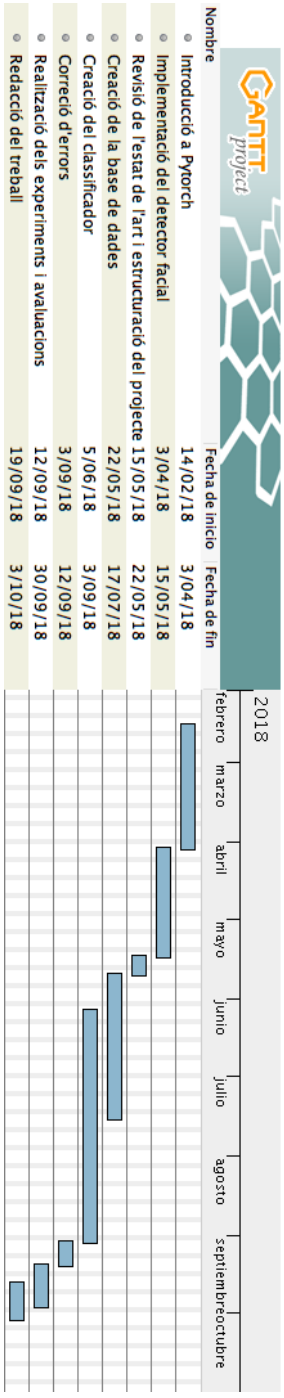


Figura 1. Diagrama de Gantt

2. Estat de l'art

El primer sistema de reconeixement facial semiautomàtic va ser desenvolupat per Woodrow W. Bledsoe [4] sota contracte del Govern dels Estats Units. Aquest sistema extreia les coordenades de punts característics de la cara, com per exemple, el centre de les pupil·les, la cantonada exterior dels ulls, etc. A partir d'aquestes coordenades es calculava una llista de 20 distàncies: l'ample de la boca, l'ample dels ulls etc.
Es podien processar unes 40 imatges per hora. El nom de la persona de la fotografia s'associava a la llista d'aquestes distàncies i s'emmagatzemava. Per tant, per poder reconèixer una persona, es calculaven aquestes distàncies i es comparaven amb el registre emmagatzemat anteriorment.

En els següents apartats explicaré com ha evolucionat cada part que he treballat en aquest projecte.

2.1 Identificació de persones no supervisada en vídeos

Generalment per la identificació de persones en vídeos s'utilitza un sistema supervisat, és a dir, prèviament es crea i s'entrena un model per cada persona que es voldrà identificar en un futur. Aquest model està format per un conjunt d'imatges que serviran per l'entrenament.

Quan no es disposa d'unes dades etiquetades prèvies, parlem d'un sistema no supervisat. Aquests sistemes solen ser multimodals, és a dir, s'utilitza la informació que ens aporta el vídeo per poder identificar el nom de les persones que hi apareixen, per exemple, els noms escrits a la pantalla, característiques facials de les persones que hi apareixen i l'àudio del vídeo.

Els mètodes actuals que s'utilitzen per a la identificació no supervisada de persones, a partir de les característiques facials, utilitzen la correlació entre els moviments de la boca i l'àudio corresponent en aquell instant de temps. Tenint en compte que només hi parla una persona i que aquesta està en posició frontal a la càmera. [5]

Per aconseguir-ho, prèviament s'anota l'àudio i el vídeo i s'entrena aquesta correlació audiovisual amb una xarxa neuronal específica: *Time Delay Neural Network* (TDNN) [6] .

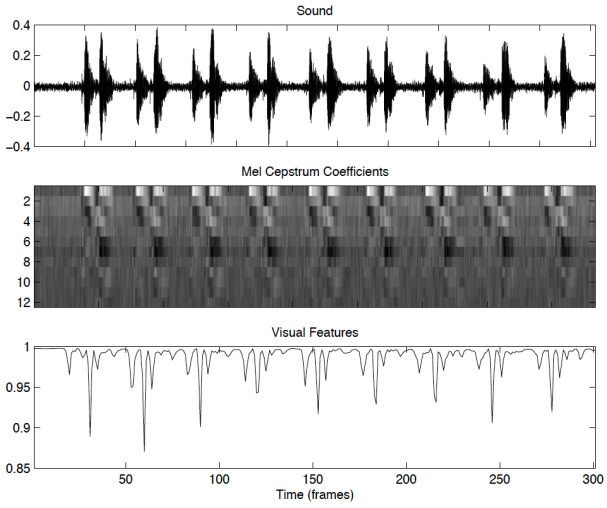


Figura 2. Exemple d'entrenament d'una persona dient "computer" nou vegades. Imatge extreta de [5]

2.2 Detecció facial

Una altre part molt important per obtenir una identificació correcte, és aconseguir un bon seguiment seqüencial de la persona etiquetada en el vídeo, així doncs, durant un seguit de imatges poder detectar la mateixa persona i etiquetar-la correctament.

La recerca de Viola i Jones [7] va permetre detectar ràpidament les cares en temps real amb una complexitat computacional molt baixa. En aquest cas, es va introduir una nova representació de la imatge anomenada *Integral Image* que permetia executar les característiques del detector molt ràpidament. D'aquesta manera, s'introduïa l'algoritme d'aprenentatge basat en *AdaBoost* [8], que selecciona petites característiques visuals i produeix classificadors més eficients. Per últim, es combinaven classificadors en cascada que permetien descartar les regions de fons.

Aquest algoritme, però, només detectava cares frontals. Tot i així al 2003, Viola i Jones van presentar una variant millorada que permetia detectar cares de perfil o rotades [9]. En aquesta nova versió, es construïen diferents detectors i classificadors per a tots els angles de la cara.

El detector de rostres més emprat en l'actualitat és el que utilitza la tècnica Histogrames d'Orientació del Gradient (HOG) [10] combinat amb un classificador lineal (*Support Vector Machine, SVM*) [11] de la llibreria *scikt-learn* [12]. Aquest detector pertany a la llibreria *Dlib*. [13] [14].

Support Vector Machine és una tècnica formada per un conjunt de mètodes d'aprenentatge supervisats. Construeix un híper-pla o un conjunt de híper-plans en un espai de dimensió infinita que es pot utilitzar per a la classificació, regressió o altres tasques.

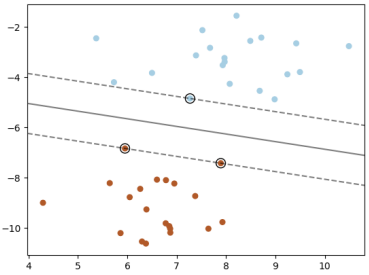


Figura 3. Representació de Support Vector Machine. Imatge extreta de [11]

Dlib és una llibreria de codi obert que té com a objectiu proporcionar un entorn de desenvolupament per l'aprenentatge automàtic. La gran diferència entre aquesta llibreria i altres existents, és que moltes d'elles no es centren en proporcionar un entorn de programari d'aprenentatge automàtic en el llenguatge C++, com ho fa *Dlib*, sinó que ho fan amb *Matlab*, *Python* o *Lua*.

En aquest treball es comparen els dos mètodes diferents de la llibreria *Dlib*: detector facial basat en HOG+SVM i detector facial basat en una *CNN* (*Convolutional Neural Network*).

2.2.1 Detector facial basat en HOG+SVM

Com s'ha comentat anteriorment, el primer detector facial de la llibreria *Dlib*, és la combinació entre els descriptors de HOG i un classificador lineal.

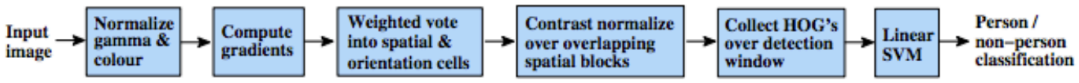


Figura 4. Visió general de l'extracció de característiques. Imatge extreta de [10]

HOG es un descriptor de la imatge que utilitza el gradient per cada píxel de la imatge per a extreure informació bàsica. A partir de la distribució d'aquests gradients es pot caracteritzar l'aparença d'un objecte o , en aquest cas, una cara. La informació local del gradient, a nivell de cadascun dels píxels, es pot agregar en forma d'histogrames calculats en diferents àrees de la imatge.

En primer lloc, es divideix la imatge en diferents cel·les i, per a cadascuna, es calcula l'histograma de les orientacions dels gradients en aquella cel·la.
Seguidament, es calculen els histogrames a totes les cel·les de la imatge i es combinen per obtenir la representació global de la imatge en forma de vector de característiques.

Tanmateix, per obtenir el descriptor, s'agrupen i es normalitzen aquests histogrames en forma de blocs. A la pràctica, aquests blocs es defineixen de forma que tinguin un solapament.

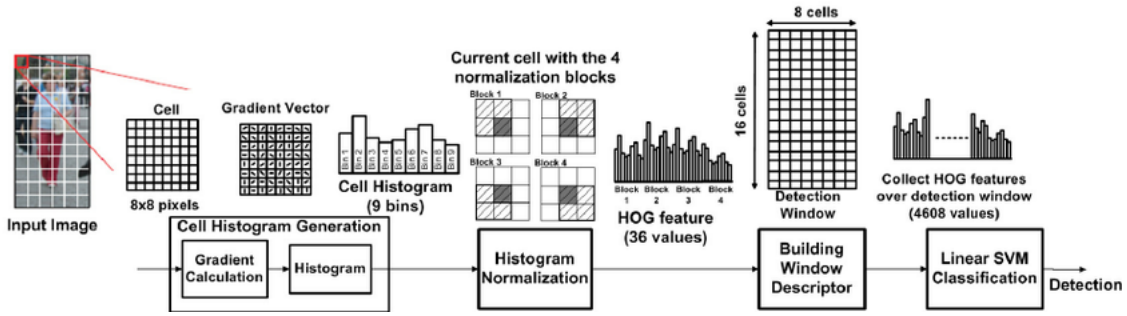


Figura 5. Extracció dels descriptors HOG per a la detecció d'objectes. Imatge extreta de [25]

Així doncs, per obtenir el detector final, primer es seleccionen unes mostres positives (allò que volem detectar) i unes mostres negatives (imatges que no contenen res del que volem detectar). A partir d'aquí, s'extreuen les característiques HOG d'aquestes mostres i s'entrena el classificador lineal (SVM). Per tant, aquest detector facial, es pot utilitzar també per a detectar objectes com podem veure a [15] [16].

2.2.2 Detector facial basat en una CNN

El detector facial basat en CNN [13] neix com a conseqüència de diversos problemes en la detecció facial utilitzant el detector basat en HOG i SVM. Entre aquestes dificultats es trobaven els canvis de posició de la cara, excés o manca d'il·luminació, expressions exagerades, etc.
Actualment, s'obtenen molts bons resultats utilitzant el detector CNN com podem veure a [17] [18] on es combinen mètodes per a calibrar el fons i poder classificar si a una regió hi ha cares o no.

Un dels principals desavantatges és que requereix un poder de computació molt més gran per funcionar, i està destinat a ser executat en una GPU per aconseguir una velocitat raonable.

3. Desenvolupament del projecte

Com hem descrit anteriorment, aquest projecte té l'objectiu de millorar la detecció facial a partir de la llibreria *Dlib* i a més a més, implementar un classificador per tal d'identificar si una persona en una seqüència de vídeo està parlant o no. En la *figura 4* podem veure com s'estructura aquest projecte:

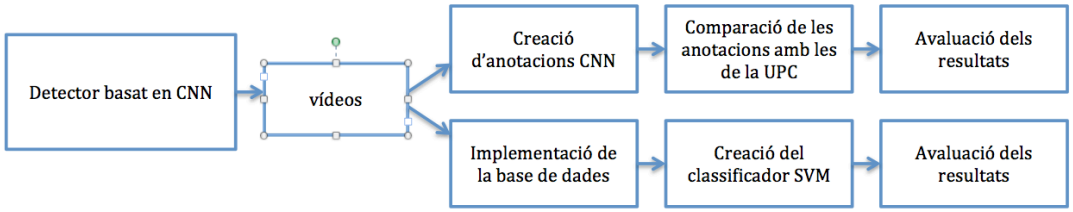


Figura 6. Estructura general del projecte.

3.1 Detector facial basat en CNN

En els últims projectes, la UPC ha utilitzat el detector facial de la llibreria *Dlib* [14] basat en HOG+SVM amb molts bons resultat. Igualment, *Dlib* defensa el detector basat en CNN per la seva bona precisió, però requereix un gran poder de computació, i això implica utilitzar GPU per obtenir una velocitat raonable. Aquest nou detector, carrega un model pre-entrenat [19] per tal de poder detectar cares en les imatges (trames).

Per efectuar una comparació coherent entre els dos detectors, s'han aprofitat les anotacions i els vídeos proporcionats per la UPC de l'any que van participar a *MediaEval 2016* [3].

La realització de les anotacions s'implementa de la següent manera: per cada vídeo, es crea un fitxer anotant el número de trama on s'ha detectat una cara i les dimensions d'aquesta. He aplicat la mateixa metodologia però, en aquest cas, canviant el tipus de detector. A partir d'aquí, manualment, s'han pogut fer dos petits estudis per determinar si el nou detector funciona millor.

El primer estudi consisteix en comprar el número de trames detectades correctament per els diferents detectors (*Dlib* (HOG + SVM) i *Dlib* CNN), cada 20 trames en un vídeo on hi apareix una cara, tal i com es mostra a la figura 6.

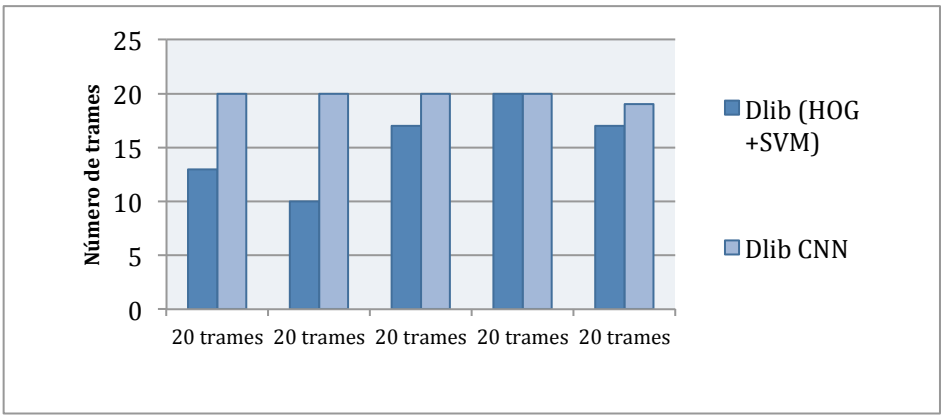


Figura 7. Diferència de cares detectades entre detectors cada 20 trames.

En el segon estudi, s'ha contat manualment el número de cares que apareixen en el vídeo d842536c-7da2-b273-cf6c-ac86c89597b9.mp4 de la carpeta "/projects/camomile/mediaeval2016/DW/DW-news-DE" del servidor de la UPC. Finalment, s'ha analitzat quantes cares ha detectat cada detector i quants falsos positius hi apareixen [Taula 2.]

Detector	Número de cares en total	Número de cares detectades	Número de falsos positius	Precisió (%)
HOG + SVM	52	35	5	67,7
CNN	52	48	1	92,3

Taula 2. Comparació de cares detectades entre el detector basat en HOG+SVM i CNN.

Per a la realització d'aquest estudi, s'han utilitzat els següents vídeos amb els seus respectius fitxers d'anotació:

- 0121200000DVBT6x1.mp4
- d842536c-7da2-b273-cf6c-ac86c89597b9.mp4
- 0122160000DVBT5x3

A continuació, és mostren algunes diferencies visuals entre els dos detectors en un instant de temps. Es pot comprovar que, per exemple, en una trama on hi apareixen 5 cares, el detector basat en HOG+SVM només en detecta una [Fig.8], mentre que el detector basat en CNN en detecta tres [Fig.9].

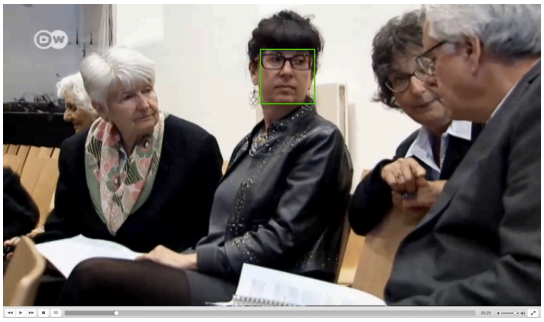


Figura 8. Resultat després d'aplicar el detector basat en HOG i SVM en una trama amb 5 cares

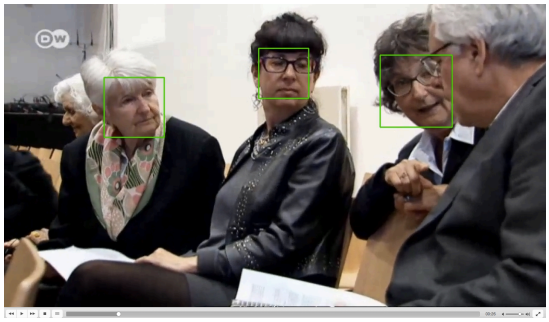


Figura 9. Resultat després d'aplicar el detector basat en CNN en una trama amb 5 cares

La gran dificultat apareix quan la posició de la cara no és exactament frontal. En les següents imatges es pot contemplar la diferència: El detector CNN detecta perfectament la cara en posició lateral [Fig.11], mentre que l'altre (HOG+SVM) no [Fig.10].

Figura 10. Resultat després d'aplicar el detector basat en HOG i SVM en una cara no frontal.

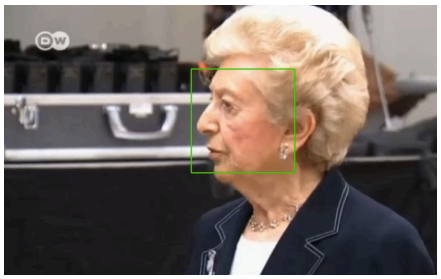


Figura 11. Resultat després d'aplicar el detector basat en CNN en una cara no frontal.

Podem deduir doncs, que el detector CNN te una precisió millor respecta l'altre en posicions de la cara no frontals. Tenint en compte aquest resultats, vaig decidir utilitzar aquest detector per a la creació de la base de dades d'aquest projecte.

3.2 Implementació de la base de dades

Abans de crear el classificador, primer s'ha generat una base de dades de cares estructurada de la següent manera:

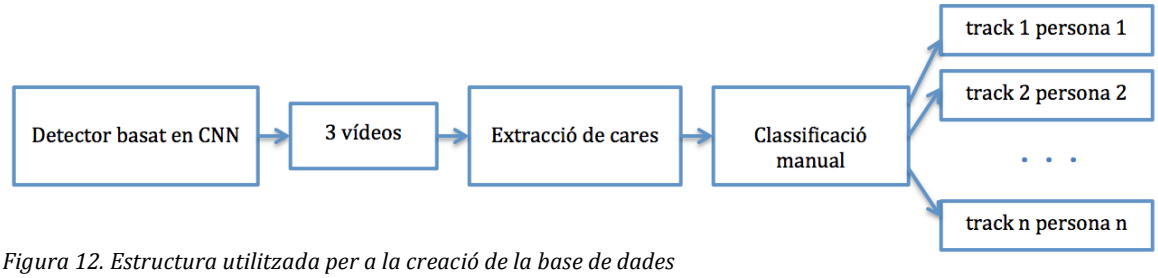


Figura 12. Estructura utilitzada per a la creació de la base de dades

Per preparar aquesta base de dades s'han utilitzat els vídeos facilitats per la UPC i d'altres extrets d'internet amb més bona qualitat:

- Els vídeos proporcionats per la UPC:
    - 18 vídeos del canal 3-24 amb una durada aproximada de hora i mitja
    - 173 vídeos del canal Alemany DW amb una durada aproximada de mitja hora
    - 100 vídeos del canal INA amb una durada aproximada de x
- Vídeos extrets d'internet:
    - Vídeo del concurs "Pasapalabra" amb una durada de
    - 4 vídeos de notícies amb una durada aproximada de hora i mitja

Primerament, s'ha passat el detector basat en CNN per tots els vídeos mencionats anteriorment. Per cada trama on s'ha detectat una cara, aquesta ha estat guardada en una carpeta. Així doncs com a resultat, tenim una carpeta amb totes les cares que apareixen en totes les trames del vídeo.

Tenint en compte que el video *tracking* és el procés de localitzar, en un conjunt de trames d'un vídeo, un mateix objecte/persona[Fig.12], un *track* és aquest conjunt de trames que pertanyen a una mateixa persona.

Un cop extretes totes les cares dels vídeos amb el detector CNN, visualment, es podia veure cada *track* per persona. Per tant, es va classificar manualment cadascun dels *tracks* en funció de si les persones parlaven (Parlen_n) o no (No Parlen_n).

Figura 13. Exemple d'un track

La base de dades consta de: 80 carpetes de persones que parlen i 115 carpetes de persones que no parlen. El nombre d'imatges a cada carpeta és variat.

A continuació es mostra l'estructura final que té la base de dades.

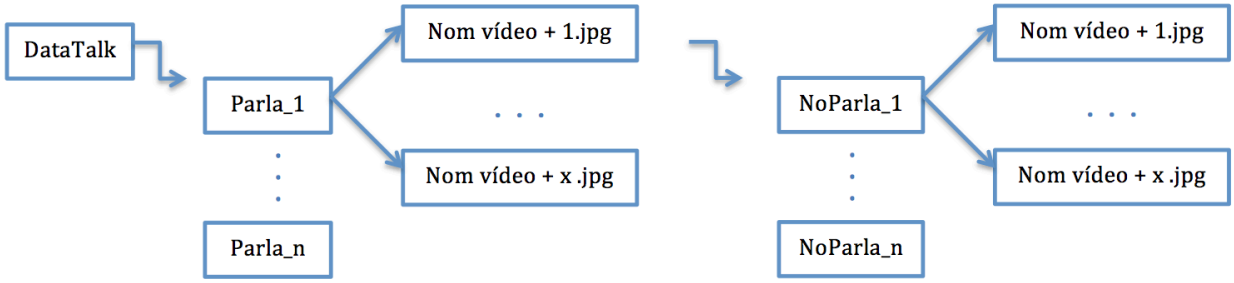


Figura 14. Estructuració de la base de dades per poder crear el classificador

3.3 Creació del classificador

Un cop acabada la base de dades de les cares, es comença a crear el classificador.

Per poder dur a terme el classificador, en primer lloc, es recorre tota la base de dades creada i, per cada imatge, es detecten els punts d'interès de la cara, dels quals, es calculen les distàncies de la boca (entre el llavi superior i inferior) i s'emmagatzemen en vectors. És a dir, per cada conjunt de imatges, s'obtenen vectors de distàncies, de la mateixa longitud.

A la hora de crear el classificador, es podria fer sobre una imatge o sobre un conjunt d'imatges en un interval de temps. En aquest cas, s'utilitza la segona opció perquè per poder predir si una persona parla, ens focalitzem en el moviment dels llavis en un interval de temps i això, ens porta a crear un vector temporal de distàncies.

3.3.1 Càlcul de distàncies

La llibreria *Dlib*, té un mètode ja creat per a la detecció de punts d'interès de la cara, anomenats *landmarks* [20]. Aquest detector pre-entrenat localitza 68 punts facials, dels quals, ens interessen els que pertanyen a la boca: del punt 49 al punt 68. A la següent figura [Fig. 15], podem veure el conjunt de punts facials detectats per *Dlib*:

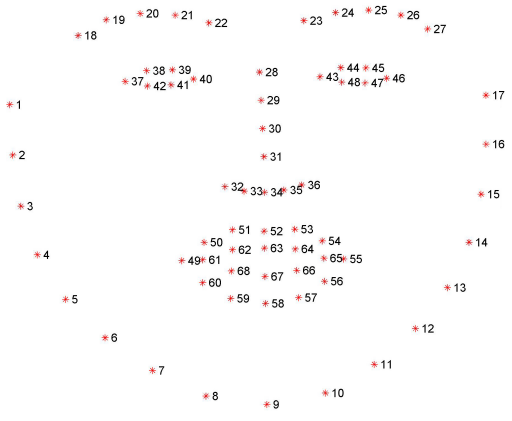


Figura 15. Representació dels punts d'interès de la cara d'un model pre-entrenat . Imatge extreta de [26]

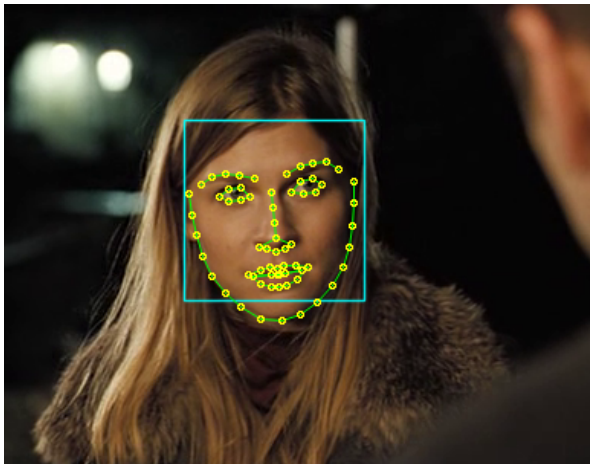


Figura 16. Aplicació del detector de landmarks a una imatge d'una cara. Imatge extreta de [28]

Per poder identificar si una persona està parlant, però, és necessari centrar-se amb els punts de la boca que proporcionen més moviment a la hora de la parla: 62, 63, 64, 66, 67 i 68, 49 i 55.

Per cada cara detectada, primer es calcula la distància *Euclidean* [21] dels punts oposats verticals (del 62 al 68) i a continuació, dels punts horitzontals (49 i 55) .

$$d_E(P,Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}.$$

On P = (p1, p2 ... , pn) i Q = (q1, q2 ... , qn) són els punts des dels quals es vol calcular la distància.

Un cop calculades les distàncies, he emparat el *Mouth Aspect Ratio (MAR)*, un mètode adaptat del *Eye Aspect Ratio* (EAR) aplicat als punts de referència de la boca:

$$\text{MAR} = \frac{\|p2 - p8\| + \|p3 - p7\| + \|p4 - p6\|}{2\|p1 - p5\|},$$

On p1,...p8 són els punts facials d'interès, descrits a la figura 15. El numerador calcula les distàncies verticals de la boca i el denominador, la distància horitzontal.

He utilitzat aquesta metodologia (MAR), ja que s'han obtingut molts bons resultats en la identificació del parpelleig d'una persona utilitzant EAR [22] .

La distància resultant quan la boca està oberta és un valor constant però, tendeix ràpidament a cero quan es tanca la boca. D'aquesta manera, és més fàcil identificar quan els llavis de la boca tenen un moviment d'obrir i tancar.

3.3.2 Generació de vectors

El meu classificador està compost de vectors de distàncies. Aquests vectors són temporals, és a dir, un vector esta format per una petita part d'un *track* d'un vídeo. Això ens ajuda a poder determinar el moviment seqüencial que té una boca al llarg del temps, i a partir d'aquí, determinar si una persona en una seqüència de trames està parlant o no.

Aquests vectors tenen una durada fixa de N posicions i cada posició d'aquest vector és la distància dels llavis d'una de les imatges classificades manualment.

Per crear els vectors, es recorre cadascuna de les carpetes amb imatges, i amb un solapament del 50% d'aquestes es van calculant les distàncies, amb el seu respectiu MAR, de les boques detectades fins a N imatges. Aquest conjunt de distàncies correspondran a un vector. Per tant, tenim vectors del següent estil [Fig. 18]. A [Fig. 17] es mostra un exemple de la creació dels vectors.

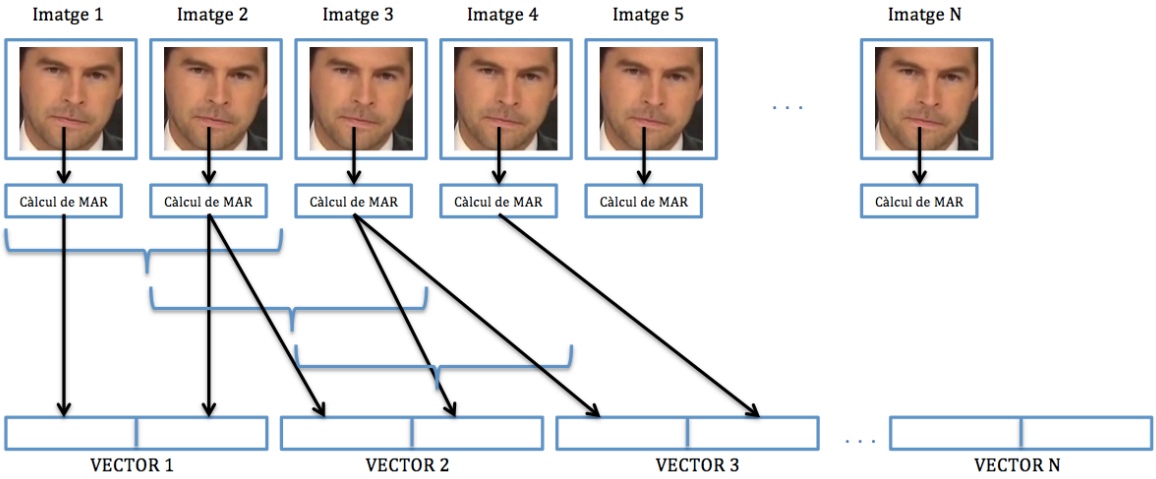


Figura 17. Exemple de la creació de vectors, en aquests cas, vectors de longitud 2.

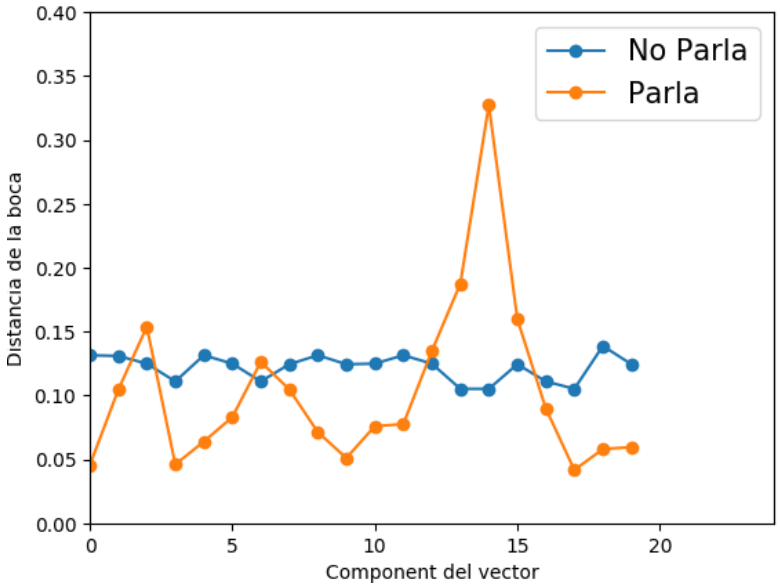


Figura 18. Representació de dos vectors de diferents classes

Podem observar la diferència entre el vector d'una persona que parla (taronja) a l'estabilitat de la que no (blau).
Aquests conjunt de vectors són emmagatzemats a un diccionari segons si aquests pertanyen al grup de 'Parla' o al grup de 'No Parla'. Finalment, disposem d'un diccionari binari ('Parla', 'No Parla') amb una llista de vectors de distàncies.

3.3.3 Creació del classificador utilitzant SVM

Per acabar, es llegeix aquets diccionari i es parteixen els vectors de distàncies en dades d'entrenament, de validació i de test. Per implementar aquesta divisió, he utilitzat el suport de la llibreria [12]. Primer, es divideixen les dades en dos grups: el 90% de les dades per entrenament i validació i el 10% de les dades per el test final, posant una llavor per tal que les dades de test sempre siguin les mateixes. Després, s'agafa aquest 90% i es torna a dividir, aquest cop però, el 80% per les dades d'entrenament i el 20% per les dades de validació. Les dades de validació, ens permeten poder modificar el tipus de classificador que es vol crear per tal d'obtenir la màxima precisió.

Per tant, en total tenim 858 vectors de 20 distàncies diferents, dels quals, 616 vectors són utilitzats per l'entrenament, 156 per les dades de validació i 86 per la realització del test final. Per a cada cas les classes estan distribuïdes equitativament.

En aquests projecte s'ha utilitzat la classe *Support Vector Classification (SVC)* de SVM per a la creació del classificador binari. Aquest classificador busca un hiperplà per separar les mostres positives de les negatives.

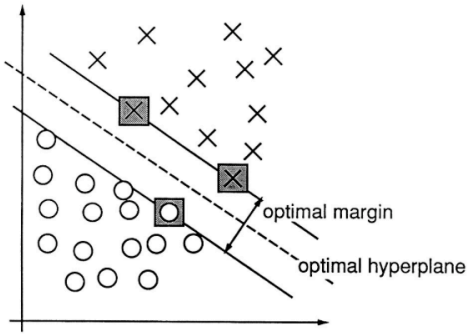


Figura 19. Exemple de separabilitat entre dues classes en un hiperplà. Imatge extreta de [29]

Donats uns vectors d'entrenament $x_i \in \mathbb{R}^p$, i = 1,...n, en dos classes, i un vector d'etiquetes $y \in \{1, -1\}^n$, SVC resol el següent problema:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i$$
$$\text{subject to } y_i (w^T \phi(x_i) + b) \geq 1 - \zeta_i,$$
$$\zeta_i \geq 0, i = 1, \ldots, n$$

on C > 0 és el límit superior, Q és una matriu positiva de n x n, $Q_{ij} \equiv y_i y_j \mathrm{K}(x_i x_j)$, on $\mathrm{K}(x_i x_j) = \Phi(x_i)^T \Phi(x_i)$ és el *kernel*, amb un hiperplà: $\langle w, x \rangle + b = 0$, on w són els pesos, b és igual a *bias* i estan condicionats per: $\min_i |\langle w, x \rangle + b| = 1$. ζ és el nombre més petit sense ser negatiu que satisfà:

$$\zeta_i = \max(0, 1 - y_i(w \cdot x_i - b)) \qquad y_i(w \cdot x_i - b) \geq 1 - \zeta_i.$$

Els mètodes *kernel*, són un conjunt d'algoritmes que porten les dades proporcionades a un espai vectorial de dimensió molt més alta que la dels vectors originals, on esperem que les classes siguin separables mitjançant un hiperplà per poder aplicar mètodes i identificar patrons. A continuació es mostren uns exemples dels diferents *kernels*, extrets de [12].

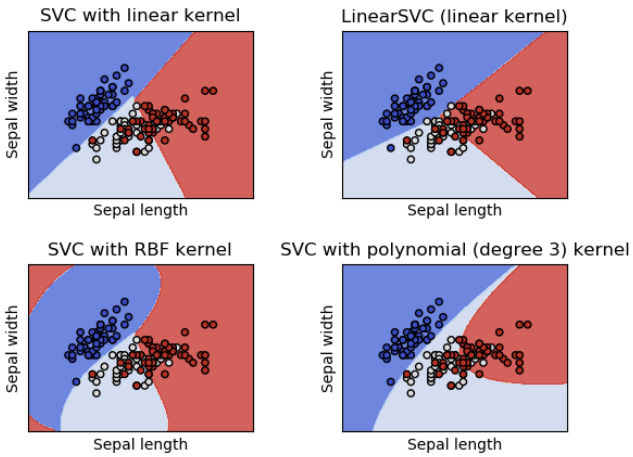


Figura 20. Exemple dels diferents kernels

Existeixen diferents tipus de classificadors depenent de quin *kernel* s'utilitzi:

- *Linear*
- *Poly*
- *Radial Basis Function (RBF)*
- *Sigmoid*

Per a realitzar aquest treball, s'ha escollit el classificador *Linear* : $\langle x, x' \rangle$ i el RBF: $\exp(-\gamma \|x - x'\|^2)$, sent γ gamma , ja que són els dos més utilitzats habitualment.

El classificador RBF té la opció de variar dos paràmetres en el moment de crear-lo, d'aquesta manera facilita adequar el classificador segons les dades d'entrenament que tenim:

- Gamma, és l'invers de la desviació estàndard del RBF. S'utilitza per mesurar la similitud entre dos punts. Un valor petit de gamma implica una funció gaussiana amb una gran variància, és a dir, es poden considerar similar dos punts encara que estiguin posicionats llunyanament. D'altra banda, un valor molt gran de gamma significaria definir una funció gaussiana amb una petita variància i, en aquets cas, dos punts es considerarien semblants només si es troben molt a prop.

- C, controla la influència de cada mostra. Si el valor de C decreix el classificador sacrifica la separabilitat lineal per guanyar estabilitat, és a dir, es comporta com un paràmetre de regularització en el SVM.

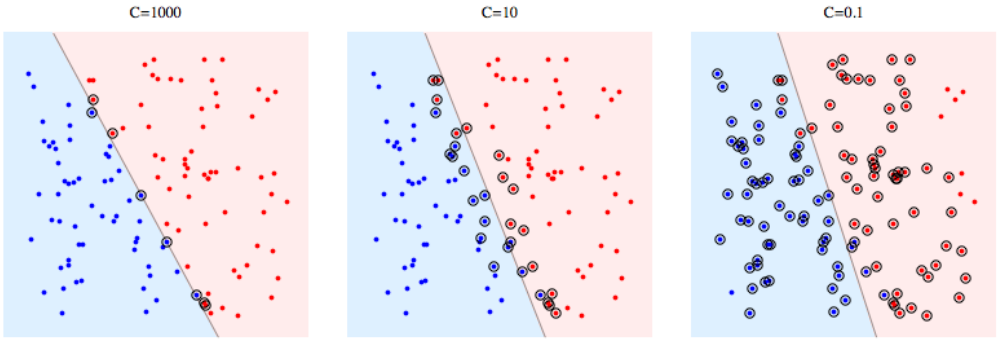
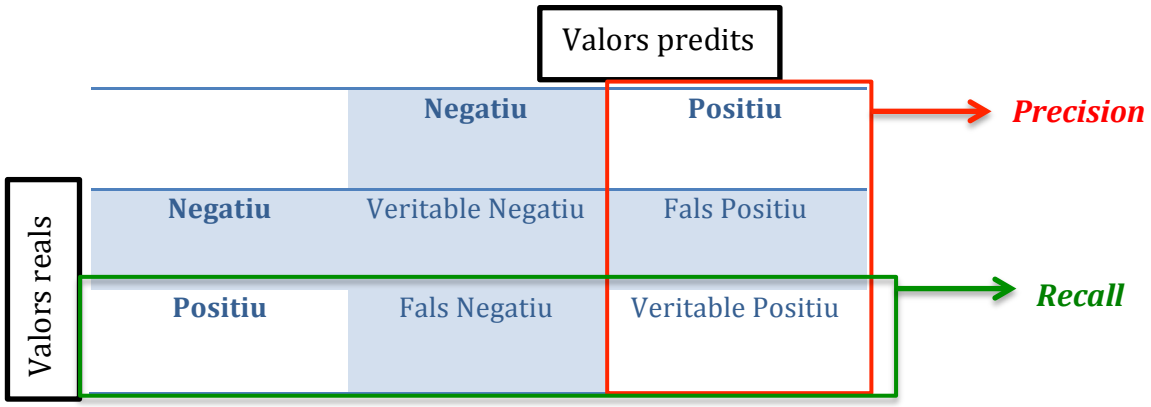


Figura 21. Diferenciació del comportament del classificador en canviar el paràmetre C. Imatge extreta de [27]

4. Avaluació dels experiments

Per la avaluació del classificador, s'ha avaluat a nivell de vectors temporals i a nivell de *track*, utilitzant la mètrica *Precision-Recall* de [12].
Precision-Recall és una mètrica que s'utilitza per analitzar models predictius. Generalment es calculen aquestes estadístiques sobre un conjunt de dades de validació i test. A continuació, explicaré amb més detall què significa *Precision* i *Recall* en aquest context.

Quan es mesura una predicció, els valors resultants es classifiquen segons si són Falsos Negatius, Falsos Positius, Veritables Negatius i Veritables Positius:



Taula 3. Diferència entre Precision i Recall respecte els valors resultants

- *Precision:* És la relació entre les prediccions correctes i les prediccions totals, és a dir, indica com de precís és el model :

$$Precision = \frac{TP}{TP+FP}$$

  On TP són els valors Veritables Positius i FP els valors Falsos Positius.

- *Recall:* És la proporció de les prediccions correctes i el nombre total de mostres correctes del model. Indica com de bo és el model mirant només les mostres reals positives :

$$Recall = \frac{TP}{TP+FN}$$

  On, TP són els valors Veritables Positius i FN els valors Falsos Negatius.

Per avaluar bé un model, es té en compte aquestes dues mètriques. Generalment quan es vol millorar la *Precision*, el *Recall* disminueix, per això, cal trobar un terme mig que proporcioni una millora considerable per al nostre model. Per això, s'utilitza la mesura *F1 score* , un equilibri de *Precision-Recall*:

$$F_1 = \left(\frac{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}}{2} \right)^{-1} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

També s'ha avaluat utilitzant la mètrica *Average Precision* de [12]. Es defineix com a la mitjana ponderada de precisions obtingudes en cada llindar de la corba de *Precision-Recall*. S'utilitza com a pes l'augment del *Recall* del llindar anterior:

$$\mathrm{AP} = \sum_n (R_n - R_{n-1}) P_n$$

On P_n i R_n són la *Precision* i *Recall* al llindar nth [23].

Així doncs, per avaluar el classificador hem realitzat dues cerques exhaustives, explicades en els apartats següents.

4.1 Variació de la longitud de vectors i el mínim de imatges analitzades:

La primera cerca es centre en el moment d'escollir la longitud dels vectors de distàncies, i el número mínim de imatges que volem que tingui la carpeta recorreguda de la base de dades. S'ha utilitzat la mètrica *Average Precision* per la avaluació, els resultats dels quals es mostren a la taula 4. Per la realització d'aquest experiment, s'ha creat un classificador SVC amb els paràmetres per defecte:

- *kernel* = RBF
- gamma = Auto, implica que gamma = 1 / número de mostres
- C = 1.

Longitud vector	Número mínim d'imatges/carpeta	AP
10	50	0,64
20	50	0,67
30	50	0,61
10	100	0,68
20	100	0,68
30	100	0,62
20	**125**	**0,69**
10	125	0,66
30	125	0,64

Taula 4. Representació dels diferents valors de AP per a cada longitud del vector i número de imatges mínimes per carpeta

Podem comprovar que la millor combinació és agafar vectors compostos de 20 distàncies amb un mínim de 125 imatges per carpeta. Aquest resultat ens fa pensar que per poder comparar si una persona està parlant, necessitem més distàncies per avaluar-ho correctament. És a dir, potser un vector de 10 distàncies és molt petit per saber si una persona està parlant, ja que, podria ser que parlés amb la boca molt tancada, però en canvi, en un vector de 20, es podria veure més la variació d'aquestes distàncies i encertar amb més precisió el estat de parla.

Pel que fa al número de imatges per carpeta, es pot remarcar que funciona millor si tenim un gran conjunt de imatges de la mateixa persona en un mateix *track* per poder efectuar la avaluació. Podríem dir que el sistema 'aprèn' millor si té un conjunt de distàncies més semblants les unes amb les altres.

4.2 Comparativa entre *kernel* RBF i *kernel* Linear

La segona cerca, ha estat comparar entre els dos *kernels*: RBF i Linear. RBF té la opció de poder variar els paràmetres C i gamma en el procés de creació, per tant, primer es calcula el valor màxim de Linear i a continuació es realitzarà una cerca exhaustiva per tal de maximitzar el *kernel* RBF.

A la taula 5 es mostren els valor de AP i *F1 score* per un classificador Linear.

Kernel	AP	F1
Linear	0,66	0,70

Taula 5. Resultat de AP i F1 score per un classificador Linear

A continuació, es va efectuar un anàlisi per determinar en quin rang de C i gamma el classificador RBF funcionava millor. A la figura 22 podem observar les diferents corbes *Precision-Recall* resultants de variar:

- gamma entre 1 i 1^{-3} i C entre 1 i 1000
- gamma entre 1 i 1^{-3} i C entre 5 i 15
- gamma entre 10 i 20 i C entre 0.1 i 10

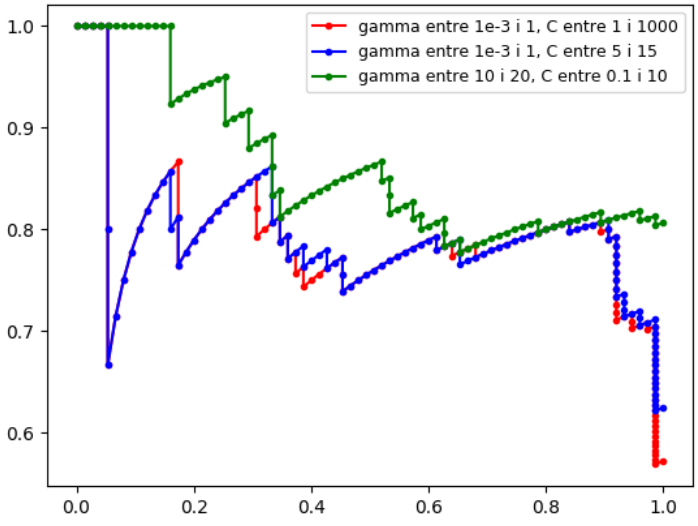


Figura 22. Curva Precision-Recall havent variat C i gamma

A partir d'aquestes variacions, s'han obtingut els corresponents F1 scores escollint el màxim valor. A la taula 6 es mostren els resultats més alts de cada combinació juntament el seu AP.

Combinació de gamma	Combinació de C	Combinació final	F1	AP
Entre 1 i $\mathbf{1^{-3}}$	Entre 1 i 1000	C = 10 Gamma = 1	0,75	0,70
Entre 1 i $\mathbf{1^{-3}}$	Entre 5 i 15	C = 6 Gamma = 1.3	0,76	0,72
Entre 10 i 20	**Entre 0.1 i 10**	**C = 0.1 Gamma = 19**	**0,85**	**0,78**

Taula 6. Representació dels valors màxims de F1 score i AP a partir de la cerca exhaustiva de C i gamma.

Comparant els resultats del dos tipus de kernels [Taula 5] i [Taula 6], he elegit per a la creació del classificador, el *kernel* RBF amb C = 0.1 i gamma = 19, ja que en aquesta combinació ens ha proporcionat el resultat de *F1 score i AP* més alt.

Utilitzant doncs, el classificador resultant, a la taula 7 es mostren els resultats utilitzant les dades de test (conjunt de vectors corresponents a parts de *tracks*):

Accuracy	Average Precision	F1
0.86	0.8	0.86

Taula 7. Resultat final del classificador aplicant les dades de test

Podem observar que els resultats són bons, ja que tenim una precisió bastant alta.

Tot i així, s'ha de comentar que per utilitzar el classificador, es necessitarà mínimament un seguit de 20 imatges (trames) on s'hagin pogut detectar correctament els *landmarks* de la boca.

4.3 Avaluació del classificador a nivell de *track*

S'ha avaluat el classificador a nivell de *track*, és a dir , donat un track (conjunt de trames d'una mateixa persona) determinar si aquella persona parla o no. Per dur a terme aquesta avaluació, només he utilitzat aquells *tracks* en que hi apareix una sola persona.

Per això, he realitzat un estudi de 50 *tracks* en total. Cada *track* és dividit per intervals de N trames temporals on cada interval és un vector de distàncies. S'han avaluat un total de 546 intervals de persones que parlen i 93 intervals de persones que no parlen, ja que, el que ens interessa principalment, és poder identificar si una persona està parlant.

Després d'analitzar cada *track*, a la taula 8 podem observar el resultat dels falsos positius, els veritables positius, falsos negatius, veritables negatius i el seu respectiu F1 *score*. Per avaluar aquests resultats, s'han tingut en compte les següents condicions:

- El resultat dels veritables/falsos positius/negatius ha estat agafant el nombre més alt de deteccions de Parla/No Parla en un *track*. Per exemple, si en un *track* hi ha una persona parlant, i el classificador ha detectat 5 vegades que parla i 1 que no, aquell *track* s'ha classificat com a Parla, i per tant, en aquest cas, com a Veritable Positiu (TP).

- En el cas que el classificador detecti el mateix nombre de cops Parla i No Parla en un mateix *track*, s'ha establert com a Fals Negatiu/Positiu, ja que, es considera que no es té clarament el resultat final.

Número total de *tracks*	Veritable Positius	Falsos Positius	Falsos Negatius	Veritables Negatius	F1 *score*
50	37	3	1	9	0,95

Taula 8. Resultat de l'avaluació del classificador a nivell de track

Podem comprovar que s'identifica correctament si una persona està parlant en un *track* amb bastanta precisió, tot i així, s'exposen situacions en els quals hem obtingut falsos positius:

- Si una persona està rient durant un seguit de temps, el classificador detecta que aquella persona està parlant. A la figura 24 es mostra un exemple:

Figura 23. Conjunt de cares predites com a parla

- Quan una persona parla i fa una pausa abans de tornar a agafar la paraula, el classificador identifica que aquella persona no està parlant. Realment, no s'hauria de tenir en compte aquest moment.

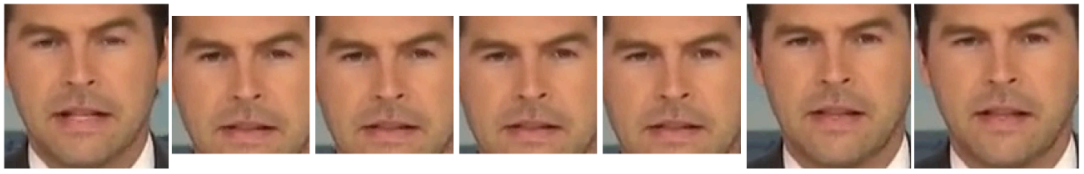


Figura 24. Conjunt de cares predites com a no parla

- Si una persona parla, però amb un moviment de la boca molt petit, el classificador identifica que aquella persona està callada:

Figura 25. Conjunt de cares predites com a no parla

Així doncs, si en un *track*, obtenim poques deteccions de No Parla entre mig o per exemple, al final de tot, classificaríem aquell *track* com a Parla. Igualment, si obtenim més quantitat de valors de Parla que de No Parla, implicaria que aquella persona està parlant.

Per acabar, He analitzat també, les dades referents als *landmarks* de la boca, de 3533 boques detectades, 1056 no s'han detectat. Per tant, si en un seguit de trames no podem obtenir les distàncies successives, ens provoca un error de precisió.

5. Pressupost

Per a la implementació d'aquest treball, s'ha establert un pressupost aproximat tenint en compte el següent:

- Les hores equivalents a la dedicació del treball, un total de 300 h. Tenint en compte que aquest treball s'ha dut a terme per un enginyer junior amb un sou de 15€/h.
- Durant el transcurs del projecte, cada setmana (33 hores en total) s'hi ha dedicat un enginyer sènior, que té un cots de 60€/h, per tant, equival a 1980€ en total
- Per el desenvolupament del projecte s'ha utilitzat un ordinador *MacBook Air,* amb un valor de 1105,6€. Però si tenim en compte que la vida útil d'un ordinador d'aquest estil és aproximadament de 3 anys, i que el projecte té una duració de 8 mesos, el cost equivalent a aquest és de 245,68 €.
- Per la execució i desenvolupament del treball, s'ha necessitat un servei addicional de computació del servidor de la UPC. Per tant, utilitzant un servei equivalent, la instància p2.xlarge dels serveis de *Amazon Web Services* (AWS) que equival a 0,78€/h. Contant 200h en total de computació, es sumen al pressupost 156€.
- També he utilitzat, aproximadament, 60 GB d'emmagatzematge al servidor, per tant a *Amazon Web Services* (AWS), l'emmagatzematge S3 estàndard és de 0,011€ per GB, equival a 0,66€.

Sou Enginyer junior	4500€
Sou Enginyer sènior	1980€
Recursos de computació	402,34€
PRESSUPOST TOTAL:	**6882,34€**

Taula 9. Pressupost total

6. Conclusions

El principal objectiu d'aquest projecte era crear un classificador per identificar si en una seqüència de vídeo, les persones que s'hi detectaven estaven parlant.

Gràcies al detector facial CNN s'ha pogut classificar un volum més extens de cares extretes de trames més seguides. Així doncs, ens ha permès tenir una precisió més acurada a la hora de computar les distàncies dels llavis de la boca.

Podem observar en els resultats que s'ha aconseguit un classificador bastant consistent. Tot i que en situacions concretes, comentades a l'apartat de resultats, podria fallar.

En un futur, es podrien desenvolupar possibles millores, com per exemple:

- Realitzar un estudi dels diferents comportaments de les persones en els vídeos de televisió i a partir d'aquí, realitzar un classificador multi-classe per identificar si una persona parla, no parla, badalla, somriu o fins i tot si plora.
- Determinar l'estat de parla, analitzant la correlació entre les dades d'àudio del vídeo i els resultats d'aquest nou classificador.
- Poder millorar la detecció dels *landmarks* de la boca. A més a més, de combinar-ho amb altres *landmarks* de la cara, com els ulls.
- Millorar les dades de classificació amb vídeos de més bona qualitat per tal de poder detectar un volum més extens de boques.

Bibliografia

[1] MediaEval Benchmark. [Online]. http://www.multimediaeval.org/about/

[2] MediaEval 2016 workshop Octobber 19–21, 2016, Hilversum, Netherlands. [Online]. http://www.multimediaeval.org/mediaeval2016/persondiscovery/

[3] India, M., Marti, G., Sayrol, E., Morros, J.R., Hernando, J., Cortillas, C., Bouritsas, G. UPC system for the 2016 MediaEval multimodal person discovery in broadcast TV task. A: Multimedia Benchmark Workshop. "MediaEval 2016 working notes proceedings". Hilversum: CEUR-WS.org, 2016, p. 1-3.

[4] Robert S. Boyer, and Larry Hines Michael Ballantyne, "Woody Bledsoe His Life and Legacy," *AI Magazine*, vol. 17, no. 1, 1996.

[5] Ross Cutler and Larry Davis, " Look who's talking: Speaker detection using video and audio correlation". In: *IEEE International Conference on Multimedia and Expo*, 2000. New York City.

[6] Daniel Povey Sanjeev Khudanpur Vijayaditya Peddinti, "A time delay neural network architecture for efficient modeling of long temporal," *Proceedings of INTERSPEECH*, 2015.

[7] Michael Jones Paul Viola, "Robust real-time face detection," *Proc. Int'l Conf. Computer Vision*, p. 747, 2001.

[8] Robert E. Schapire Yoav Freund, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *journal of computer and system sciences*, vol. 55, pp. 119-139, 1997.

[9] Michael Jones Paul Viola, "Fast Multi-view Face Detection," *Mitsubishi Electric Research Lab TR-20003-96*, vol. 3, no. 14, p. 2, July 2003.

[10] B. Triggs N. Dalal, "Histograms of oriented gradients for human detection," *IEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886-893, 2005.

[11] [Online]. http://scikit-learn.org/stable/modules/svm.html

[12] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, et al. "Scikit-learn: Machine Learning in Python," *The Journal of Machine Learning Research* , vol. 12, pp. 2825-2830, 2011.

[13] Davis E. King, "Dlib-ml: A Machine Learning Toolkit," *Journal of Machine Learning Research 10*, vol 10, pp.1755-1758, Jul 2009.

[14] Davis E. King. (202) Dlib c++ Library. [Online]. http://dlib.net

[15] Christian Wojek, Bernt Schiele, Pietro Perona Piotr Dollás, "Pedestrian Detection: A Benchmark," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.

[16] Shahraki , F. F., A. P., Regentova, E. E., & Muthukumar, V, "Bicycle Detection Using HOG, HSC and MLBP", In International Symposium on Visual Computing, pp. 554-562.,Springer International Publishing, 2015.

[17] Jiafu Wu Yicheng An, Chang Yue, "CNNs for Face Detection and Recognition". Stanford, In *CS231n: Convolutional Neural Networks for Visual Recognition*, Spring2017

[18] Zhe Lin, Xiaohui Shen, Jonathan Brandt, Gang Hua Haoxiang Li, "A Convolutional Neural Network Cascade for Face Detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, 2015.

[19] [Online]. http://dlib.net/files/mmod_human_face_detector.dat.bz2

[20] [Online]. http://dlib.net/face_landmark_detection.py.html

[21] Travis Oliphant, Pearu Peterson Eric Jones. scipy.spatial.distance.euclidean. [Online]. https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.spatial.distance.euclidean.html

[22] JanCech TerezaSoukupová´, "Real-Time Eye Blink Detection using Facial Landmarks," in *21st*

Computer Vision Winter Workshop, Rimske Toplice, Slovenia, 2016.

[23] Sklearn.metrics.average_precision_score. [Online]. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html#rcdf8f32d7f9d-1

[24] Travis Oliphant, Pearu Peterson Eric Jones. (2001--) SciPy: Open source scientific tools for Python. [Online]. http://www.scipy.org/

[25] Vivienne Sze Amr Suleiman, *Energy-Efficient HOG-based Object Detection at 1080HD 60 fps with Multi-Scale Support.*, 2014.

[26] Facial point annotations. [Online]. https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/

[27] [Online]. https://stackoverflow.com/questions/4629505/svm-hard-or-soft-margins

[28] Carmelo Marin A. (2017, Novembre) Face Landmarks Detector con Dlib y OpenCV. [Online]. http://acodigo.blogspot.com/2017/11/face-landmarks-detector-con-dlib-y.html

[29] Vladimir Vapnik Corina Cortes, "Support-Vector Networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273-297, Sept 1995.